

Unslotted Deflection Routing: A Practical and Efficient Protocol for Multihop Optical Networks

Thierry Chich, Johanne Cohen, and Pierre Fraigniaud

Abstract—This paper is concerned with all-optical networks using deflection routing and time division multiplexing. Slotted networks make use of the synchronous arrival of the packets to the routers to minimize locally the number of deflections. In this paper, we show that the difference in performances between slotted and unslotted networks is mainly due to the fact that unslotted networks cannot easily perform such local optimization. We also show that minimizing locally the number of deflections in unslotted networks gives rise to an NP-complete problem. To overcome this problem, we have designed a heuristic whose aim is to limit locally the number of deflections. We experimentally demonstrate that this heuristic enhances unslotted routing almost at the same performance level as slotted routing. As a consequence, we have shown that unslotted deflection routing can be implemented in a way which makes it a competitive alternative to slotted deflection routing for optical time division multiplexing deflection networks.

Index Terms—All-optical networks, deflection routing, slotted versus unslotted networks.

I. INTRODUCTION

ALL-OPTICAL networks provide high bandwidth and fault-tolerant communications by avoiding the bottleneck due to the electro-optic conversion. Several types of optical networks have been considered in the literature [1], [4], and several methods have been proposed to share the bandwidth of optical networks. Among them, time division multiplexing (TDM) is a technique used to improve the bandwidth of a single wavelength channel [23]. In TDM networks, every message is decomposed in packets that are routed independently. Several studies (including experimental test-beds) have been carried out to produce high-capacity optical packet routers [13], [22]. Deflection routing is a frequently proposed protocol in this context, in particular because it does not require large buffers [4], [6], [19]. The main characteristic of deflection routing is that a packet requesting a busy output link, that is a link

already used by another packet, is deflected on a free output link. This technique allows to avoid packets destructions inside the network, and therefore simplifies the management of the network.

Most of the papers in the literature assume slotted systems for optical TDM (OTDM) networks [2], [9], [25]. In such systems, each packet is inserted in a time-slot of fixed duration. A time-slot includes the header and the payload, which are conveyed at different bit rates. All incoming slots entering a router on different input links are synchronized.

Slotted systems offer many advantages. For instance, packets can be inserted in the network as soon as a free time-slot is available. Also, slot synchronism allows to locally optimize the requests of the packets, and therefore to limit the number of deflections. Additionally, synchronous routing allows the use of rearrangeable multistage switches to pipeline the switching. More generally, see, e.g., [11], [16] for discussions about the way to improve optical packet-switching. However, slotted systems also present some drawbacks. For instance, slotted systems require additional hardware that produces important degradation of the signal [23]. Also, fixed slot length does not allow to adapt the packet-size to the need of the application. Finally, slotted networks are very sensitive to faults in the synchronization system. This is why unslotted systems were proposed as an alternative for OTDM networks [7], [8], [10].

On the positive side, unslotted systems do not require synchronization hardware (but a standard non blocking switch). Moreover, packet-sizes can be tuned according to the needs of the users, i.e., each packet has a length proportional to the size of the data that it conveys. However, it is often *claimed* that unslotted systems present three major drawbacks:

- 1) Unslotted systems may cause important congestion phenomena [7], [8];
- 2) Unslotted systems do not succeed to make use of the whole bandwidth of the network because of the variability of the interpacket spaces; and
- 3) Asynchronous packet arrivals do not allow to maximize the global satisfaction of the packets by minimizing locally the number of deflections.

In this paper, we show that these three problems are either of minor influence, or can be overcome. More precisely:

- 1) We show that the bandwidth lost by interpacket spaces in unslotted networks is not more significant than the bandwidth lost by small packets requiring individual slots in slotted networks.
- 2) We show that, in unslotted networks, congestion appears only for specific configurations of the network, and thus it can be easily avoided.

Manuscript received March 27, 1998; revised July 9, 2000; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Sivarajan. A preliminary version of this work was presented at the IEEE Globecom, November 1998. This work was supported in part by NATO and Carleton University. The work of T. Chich was supported by the Centre National d'Etude des Télécommunications (CNET) under Grant 9408B54. The work of J. Cohen was supported by the Direction des Etudes Techniques (DRET) of the DGA. The work of P. Fraigniaud was supported by the RNRT program R.O.M. (Réseaux Optiques Multiservices).

T. Chich is with the Centre Universitaire des Sciences et Techniques, 63170 Aubière, France (e-mail: chich@garp.univ-bpclermont.fr).

J. Cohen is with LORIA, Campus Scientifique, 54506 Vandoeuvre les Nancy, France (e-mail: Johanne.Cohen@loria.fr).

P. Fraigniaud is with the Laboratoire de Recherche en Informatique—CNRS, University of Paris-Sud, 91405 Orsay Cedex, France (e-mail: pierre@lri.fr).

Publisher Item Identifier S 1063-6692(01)01314-0.

- 3) The third problem, that is the local optimization of the packet-to-link requests, is actually the most important problem. In particular, it cannot be solved via small hardware adjustments, but requires an algorithmic solution. Unfortunately, we show that the underlying problem corresponding to optimizing the packet-to-link requests is NP-complete in the context of unslotted networks. Nevertheless, we also show that it is possible to derive a fast and efficient heuristic for that problem. This heuristic allows to increase the throughput of unslotted networks of about 35%.

As a consequence, unslotted deflection routing is a competitive alternative to slotted deflection routing for optical TDM networks.

Structure of the paper. The next section gives preliminaries about deflection routing in all-optical networks. Section III precisely describes our experimental protocol for comparing slotted and unslotted networks. Section IV compares slotted and unslotted networks, and shows that the two first claimed drawbacks of unslotted networks are not critical. Section V presents our heuristic which locally limits the number of deflections in unslotted networks. Section VI gives the formal proof that minimizing locally the number of deflections yields an NP-complete problem. Finally, Section VII contains some concluding remarks.

II. OPTICAL TDM NETWORKS

This section is devoted to a brief description of the main characteristics of optical TDM networks, in particular the distinction between slotted and unslotted networks.

A. Routing in All-Optical Networks

In all-optical TDM networks, a packet is composed of its payload and its header. The payload contains the data (files, images, sounds, etc.), and the header includes useful information for the routing function (destination label, packet number, source label, etc.). The payload circulates at the photonic rate whereas the bandwidth allocated to headers is limited by the electronic bottleneck. When a packet arrives at a given router, its header is converted in electronic format, and it is decoded by the routing control processor (RCP) which takes the routing decision. Once the routing decision has been taken, that is once a single output port has been selected, the router connects the input port of the packet to this output port so that the payload can cut through the photonic switch. The payload is possibly slightly delayed in a loop while the RCP is performing its computation. Finally, the RCP generates a new header which is added to the outgoing payload.

The routing decision at node x , that is on which output link must be routed a packet entering x , is usually taken according to a routing table T_x . The entry $T_x[i, y]$ of T_x is set as a measure of the “quality” of routing a packet currently in x , and of destination y , through the output link i . For instance,

$$\begin{cases} T_x[i, y] = 1 & \text{if the link } i \text{ is on a shortest path} \\ & \text{from } x \text{ to } y; \\ T_x[i, y] = 0 & \text{otherwise,} \end{cases}$$

specifies a routing in which packets are routed along minimal paths. This routing can be adapted by giving *preferences* to one or more specific shortest paths (e.g., the Z^2 -routing [3] on a mesh or torus gives preference to the shortest path(s) that route the packet closer to the diagonal between the source and the destination). For instance, on a 4×4 router, $T_x[1, y] = 2$, $T_x[2, y] = 1$, $T_x[3, y] = 0.2$, and $T_x[4, y] = 0$, can be interpreted as follows: a packet entering x , and destined to y , should be routed on link 1 or 2, with a preference to link 1, and, if both links are busy, then it should be deflected either on link 3 or 4, with a preference to link 3. Usually, preferences are normalized so that they can be fairly compared. This would yield: $T_x[1, y] = 5/8$, $T_x[2, y] = 5/16$, $T_x[3, y] = 1/6$, and $T_x[4, y] = 0$, so that the sum of the preferences is 1. Ideally, the ratio $T_x[i, y]/T_x[j, y]$ should reflect the ratio of the expected number of hops to reach the destination y using link i or link j .

The design of the routing tables T_x 's, for all nodes x of the network, is not considered in this paper. The purpose of this paper is the optimization of the routing in case of contention between packets “requiring” or “having preferences” to the same set of output links. This optimization is performed based on the packet-to-link preferences in order to maximize the global satisfactions of the packets.

B. Slotted and Unslotted Networks

Packet Size: Slotted networks impose packets of fixed length whereas unslotted networks allow to adapt the length of the packets to the amount of data there are conveying. It is somewhat difficult to choose the “optimal” packet size in slotted networks. In particular:

- 1) too short packets induce over-costs due to the reconstruction of the message from the received packets at the destination;
- 2) too long packets induce a significant waste of bandwidth.

Packet Injection: Inserting packets in slotted networks is easy: it only requires to test whether a slot is empty among the incoming slots. Insertion is a bit more complex in unslotted networks. Four solutions have been proposed in [8]. One of them requires to discard inserted packets as soon as they contend with incoming packets. Two others do not always give priority to transit packets. This suggests to adopt the fourth solution of [8]. A fiber loop is added to each input link in order to delay the arrival of the packets [see Fig. 1(a)]. The arrival times of packets entering the loop are taken into account by the RCP to decide whether there is enough space to insert a packet. This strategy is applicable as soon as packets are of bounded length. However, as opposed to the case of slotted networks, this system does not change the fact that the links cannot be fully occupied by packets, and the interpacket space is *a priori* arbitrary.

Synchronization: Slotted networks require to synchronize the arrival time of the incoming packets because providing links of length multiple of the packet length is not enough due to temperature variation and fiber chromatic dispersion [22]. Synchronization is performed by the introduction of switchable delay lines [5], [23] [see Fig. 1(b)]. The number of traversed optical couplers is at least logarithmically proportional to

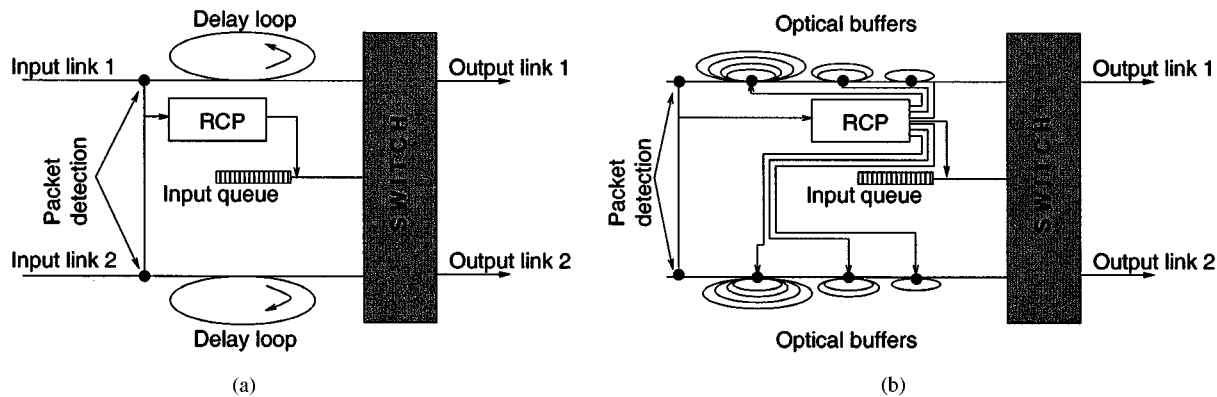


Fig. 1. (a) Delay loops in unslotted networks. (b) Synchronization systems in slotted networks.

the product of the precision of the synchronization by the maximum delay between two packets. This induces power losses and decreases the robustness of the network. Moreover, there should be enough space between two consecutive packets traversing the same link so that the first packet can be delayed for the purpose of synchronization. This significantly reduces the occupancy of the links in slotted networks.

Routing Optimization: Given k incoming packets m_1, \dots, m_k to node x , let us denote by $p_{i,j}$ the preference of packet m_i for the output link j , $j = 1, \dots, n$. In all our experiments, the preferences of a packet are precisely set up by the routing table T_x , i.e., if the packet m_i is destined to y , then $p_{i,j} = T_x[j, y]$. However, preferences could be set according to many other parameters such as priority level or alternative routing strategies (source-routing [24], centralized protocol, etc.). Therefore, in the remaining of the paper, we refer to the preferences $p_{i,j}$'s rather than to the routing tables T_x 's.

Slotted networks take benefit of the simultaneous arrival of the packets. Preferences induce a weighted complete bipartite graph whose first set of the partition represents the k packets, and the second set represents the n output links. Therefore, a natural way to optimize locally the routing in slotted networks is to compute a maximum weighted matching in this bipartite graph [17]. If the edge (i, j) belongs to the matching, the packet m_i is said to be *assigned* to the output link j . The polynomial complexity of the maximum weighted matching [15] makes this solution realistic in this context. A packet is said to be *deflected* when it is assigned to an output link which does not correspond to a shortest path between the current node and the destination.

Such optimization cannot be performed in unslotted networks because packets enter routers asynchronously, and they are therefore routed sequentially. This induces many deflections that are avoided in synchronous (slotted) systems. For instance, let m_1 and m_2 be two packets arriving at node x by two different input links, roughly at the same time. In asynchronous systems, these messages are treated separately. Assume that m_1 is treated first, and then m_2 . If m_1 has equal preferences for the output links 1 and 2, then the router will choose one of these two links arbitrarily; Say it routes m_1 through link 1. As a consequence, if m_2 has a unique preference for link 1, m_2 will be deflected. It would have been more efficient to route m_1 through link 2, and m_2 through link 1, but asynchronous

systems do not provide such optimization. In this paper, we will show how this can be improved.

III. AN EXPERIMENTAL MODEL FOR OTDM NETWORKS

To perform simulations, we have modeled a metropolitan area network (MAN), say, of size of a city. In this section, we precisely describe the topology, the link capacity, the traffic, etc., of this network.

A. Topology and Routing

We have considered the *bidirectional Manhattan street network* that is the symmetrically oriented torus, i.e., a mesh with wraparound links. The size of the torus is fixed at 12×12 . More precisely, nodes are labeled by pairs (x, y) , $0 \leq x \leq 11$ and $0 \leq y \leq 11$. Node (x, y) is connected from and to nodes $(x, y + 1 \bmod 12)$, $(x, y - 1 \bmod 12)$, $(x + 1 \bmod 12, y)$, and $(x - 1 \bmod 12, y)$. Each router is therefore supposed to be a 5×5 crossbar: one of the bidirectional link is devoted to the input-output of the optical network.

We have used the so called Z^2 shortest path routing [3]. This routing selects the output link supporting the maximum number of shortest paths from the current node to the destination. For instance, if the source and the destination are the two opposite corners of a square, then (in absence of contention), the route will zig-zag between these two corners. More formally, for any two nodes x and y , and any output link i of node x , let $N_x[i, y]$ be the number of shortest paths from x to y that pass through link i . The routing table corresponding to the Z^2 routing is defined by: $T_x[i, y] = (N_x[i, y] / \sum_j N_x[j, y])$.

We have fixed the size of the input queue at 100 packets at each node. The bandwidth of the links is supposed to be 10 Gb/s, and each link is supposed to have a length of 2 kilometers. The packet-headers are conveyed at 622 Mb/s.

B. Traffic

1) *Packet Length:* According to standard IP-traces, the length of the packets follows a bimodal law polarized at 1) the length of the acknowledgment packets, and 2) the length of the maximum packet size. Indeed, since every message is decomposed in packets by the network-application interface, a lot of packets are of length the maximum packet size. Few other packets are smaller, in particular packets corresponding

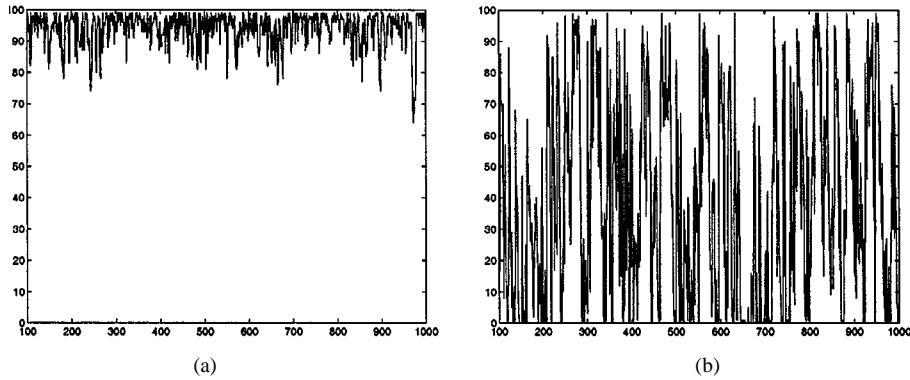


Fig. 2. (a) Poisson traffic versus (b) sporadic traffic: number of packets in a queue as a function of the time.

to the tail of messages which are not of a length multiple of the maximum packet size.

The choice of the “optimal” packet-size for deflection routing (in a fixed packet size context) has been the source of many discussions.

- 1) On one hand, a too small packet (as an ATM cell) would not be suitable because of the need of reordering packets, and because of the too large ratio header size over payload.
- 2) On the other hand, a too large packet would produce a waste of bandwidth as small messages will not fill up the slot.

In our experiments, the time slot is set to be $1 \mu\text{s}$ in slotted networks. Indeed, $1 \mu\text{s}$ at 10 Gb/s represents 10 Kb , that is roughly twice the most frequent size of an IP packet. The minimum size of a packet is 200 ns . Indeed, $622 \times 200 \text{ ns} = 15 \text{ bytes}$, that is less than the size of an IP header. Let L be the length of a packet. We set $\text{Prob}(L = 200 \text{ ns}) = 0.3$, and $\text{Prob}(L = 1 \mu\text{s}) = 0.4$. The other packet lengths are chosen as multiple of $0.1 \mu\text{s}$, uniformly in the interval $[0.3 \mu\text{s}, 0.9 \mu\text{s}]$. The average length of a packet is thus 642 ns , that is the bit load of a slot is 6.42 Kb .

Unslotted networks support packets of different lengths. We have fixed the maximum size of a packet at $1 \mu\text{s}$ (that is the size of the slot of slotted networks) in order to facilitate the comparison between slotted and unslotted networks. This equality implies that the packet length distribution is the same as previously described. In particular, the average length of a packet is 642 ns .

2) *Injection Rates*: Simulations of slotted networks are usually performed using a simulation tick exactly set to the time slot. At each tick, all the routers of the network are scanned and the routing is performed. The packet-injections are set according to a Bernoulli law. Unslotted networks could be simulated in a similar way using a shorter simulation tick. However, if the simulation tick is shortened, the average of the Bernoulli law must also be decreased in order to obtain the same average offered load as in slotted networks. The side effect would be that the injection laws of slotted and unslotted simulations would not be exactly the same. Therefore, to perform slotted and unslotted simulations in the same setting, we have separated the simulation tick and the injection tick. Precise details are given in Appendix A. Using this setting, we have performed experiments which showed that simulations performed with a tick equals to

$1/200$ of a time slot give similar results as simulations performed with a tick equals to the $1/10$ of a time slot. Therefore, in all our experiments, the simulation tick is set to $1/10$ of the time slot.

Finally, in order to simulate sporadic traffic observed in real networks [18], [21], we have modeled the packet injection law by a two states Markovian chain. The method used to obtain the same law for slotted and unslotted networks is detailed in the Appendix B. One can see in Fig. 2 that Poisson and sporadic traffics are indeed very different.

C. Experimental Measures

All measurements are performed at the steady state, on a single run of $10^6 \mu\text{s}$ (the steady state is always reached after at most $5 \cdot 10^3 \mu\text{s}$). We have measured the *throughput* of the network as a function of the input demand. More precisely, we have counted the average number of packets that arrive at destination every μs , divided by the number of nodes (that is 144). The *throughput* is in $[0, 1]$ for slotted networks. Note that the throughput per node and per μs expressed in byte can be obtained in both slotted and unslotted networks by a simple multiplication by 6.42 Kb . The input demand is the average number of packets that each node sends at each step. Since input queues are of bounded size, packets can be lost when the network approaches the saturation. The number of *lost packets* is then inversely proportional to the throughput. We have also considered the *link occupation* of the network, that is the percentage of the bandwidth used at the steady state.

Furthermore, we have created a specific traffic, called *spy traffic*, between two given nodes in order to obtain local measurements. In our experiments, node (2, 2) sends packets to node (9, 9) according to a Poisson law of mean 0.01 (i.e., at a low rate). We have reported the average number of times spy packets are deflected. For a sake of uniformity, we have normalized the results as a function of the number of received packets.

IV. COMPARISON BETWEEN SLOTTED AND UNSLOTTED NETWORKS

The aim of this section is to compare slotted and unslotted networks, and to discuss the three problems supposed to be caused by unslotted systems, and stated in the introduction: 1) congestion phenomena, 2) waste of bandwidth, and 3) large number of deflections.

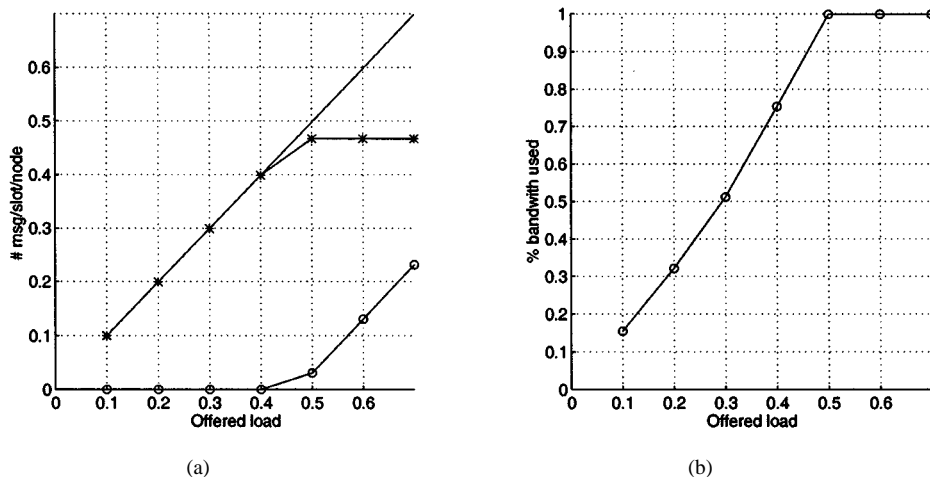


Fig. 3. Slotted routing under Poisson traffic. (a) Throughput (*) and lost packets (o). (b) Link occupation.

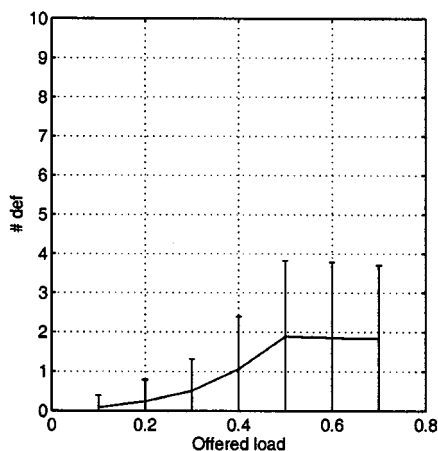


Fig. 4. Average and standard deviation of the number of deflections for slotted routing under the Poisson traffic.

A. Slotted Routing

Fig. 3 presents the well know behavior of synchronous routing under Poisson traffic. Fig. 3(a) shows the two states of the network: a linear increase of the throughput until the network gets saturated. When the network saturates, the throughput becomes constant, and the number of lost packets increases (whereas no packets are lost for a low offered load). One can check that the network starts to saturate for an offered load larger than 0.45 (either by comparison with the diagonal line, or by looking at the number of lost packets).

Fig. 3(b) presents the average number of packets per link. Again, the result is not surprising. When the offered load increases, the number of packets per slot increases more than linearly. This is due to the interactions between the network load on one hand, and the number of packet deflections on the other hand. When the network reaches the saturation, the whole bandwidth of the network is used. This is always the case for slotted networks.

Fig. 4 shows that, under low traffic condition (that is for an offered load at most 0.5), the number of deflections increases super linearly. After the saturation threshold, the number of deflections does not significantly change. The same behavior can be observed for the standard deviation.

Fig. 5(a) and (b) show the influence of a sporadic traffic on slotted routing. Fig. 5(a) shows that, when the network is not yet saturated, the number of lost packets is larger under the sporadic traffic than under the Poisson traffic. The difference looks rather small on the curves, but such a small difference corresponds to thousands of packets that are lost in case of sporadic traffic (actually many packets are lost even for an offered load of 0.1). This is due to the large standard deviation of the bi-Poisson traffic. When the network is saturated, the two types of traffic offer the same behavior. As one can check on Fig. 5(b), the loss of packets under a sporadic traffic imply that the links saturate for a larger offered load than for Poisson traffic. The number of lost packets is the major difference between Poisson and sporadic traffic. However, for a same number of packets inside the network, the behavior of these packets is roughly the same for both traffics.

We did not observed significant differences between sporadic and Poisson traffics when looking at the distribution of the latencies. Tiny improvements under the sporadic traffic come from the smaller average number of packets per slot in this mode. This confirms the fact that the internal behavior of a deflection network is roughly independent of the traffic nature.

B. Unslotted Routing

1) *Problem 1: Congestion Phenomenons:* Fig. 6(a) shows that the throughput of unslotted routing is qualitatively the same as the one of slotted networks. In particular, the saturation state is stable, that is there is no degradation of the throughput as the offered load increases. This is in contradiction with a similar study in [7], [8] which observed a severe degradation of the throughput. However, the experiments performed in [7], [8] assume packets of fixed length (although the routing is unslotted). Unslotted routing with fixed size packets induces resonance phenomenons when the length of the links is a multiple of the packet size. It is actually pointed out in [8] that “assuming a fully occupation of the links, any packet arriving at a node finds just one output link free and is forced to follow the path of its predecessor.” All these phenomenons induce livelocks that strongly reduce the throughput of the network, which is also sensible to the length of the delay loops. However,

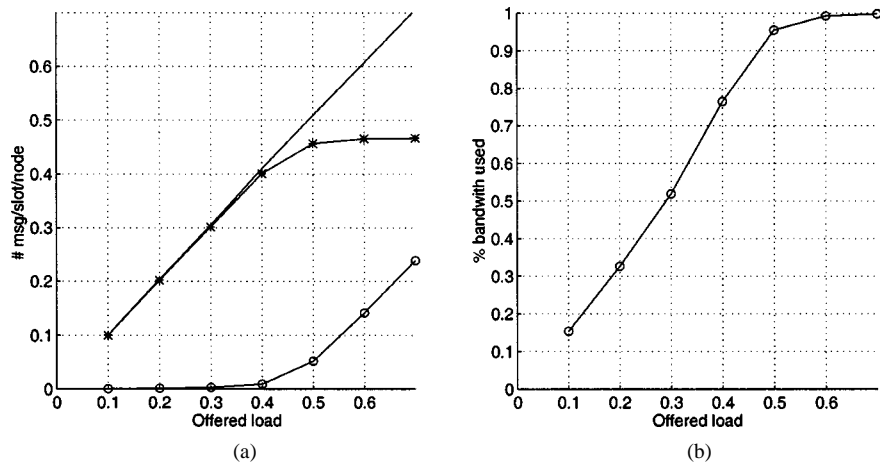


Fig. 5. Synchronous routing under sporadic traffic. (a) Throughput (*) and lost packets (o). (b) Link occupation.

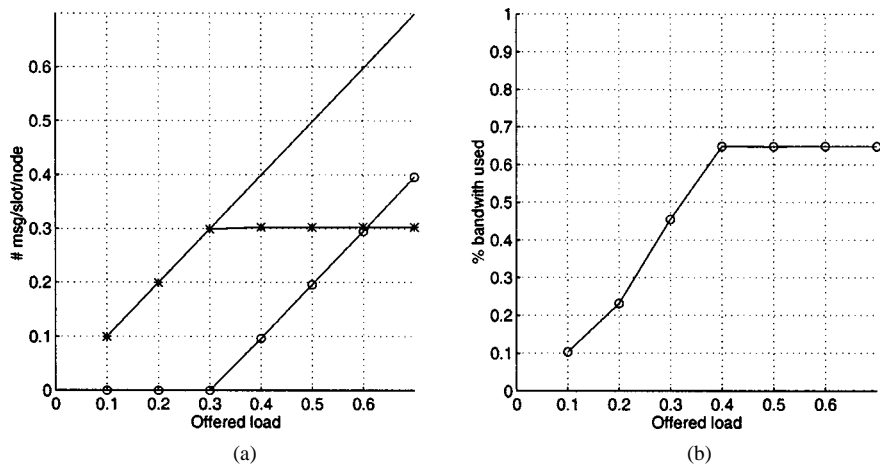


Fig. 6. Unslotted routing under poisson traffic. (a) Throughput (*) and lost packets (o). (b) Link occupation.

in standard unslotted networks, packets are of variable length and all these resonance phenomena do not occur, and there is no congestion phenomenon.

2) *Problem 2: Waste of Bandwidth:* Fig. 6(b) presents the link occupation of unslotted networks. At the saturation state, the link occupation is near 0.7. It corresponds to 1.1 packets per μs . Unslotted networks cannot totally fill up the links because too small interpacket space does not allow to insert packets (we have measured an average inter packet space of roughly $0.3 \mu s$). We did not present results on bursty traffic since, as shown in [10], there is no big difference between Poisson and sporadic traffic in unslotted networks, as far as the internal traffic is concerned.

Even if unslotted networks present qualitatively the same behavior as slotted networks, there is quantitatively a big difference. In order to understand why such a difference, we have run experiments on greedy routing in slotted networks. Greedy routing considers sequentially the packets arriving at a node in the same time slot. It assigns to the current packet the not yet assigned output link which maximizes the preference of the packet. This strategy is similar to the usual unslotted routing since it does not make use of the global preferences of the packets arriving at a node in the same time slot.

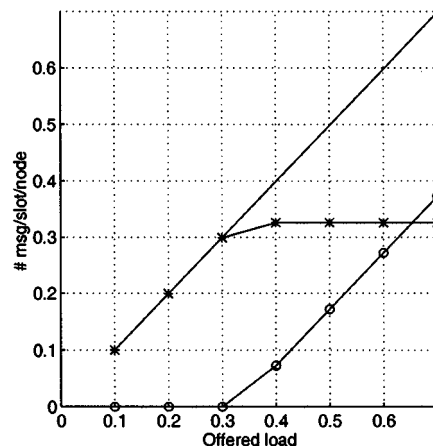


Fig. 7. Throughput (*) and number of lost packets (o) for the greedy slotted routing under Poisson traffic.

Fig. 7 presents the behavior of greedy routing under a Poisson traffic. One can notice a large degradation of the performances in comparison with synchronous routing. For instance, the network get saturated for a much smaller offered load (roughly 0.3 rather than 0.45). Thus, as observed in [12], a sequential rather than simultaneous treatment of the packets does not allow to reduce the

number of deflections locally, and the throughput decreases: the throughput of greedy slotted routing is roughly the same as the throughput of unslotted routing. Moreover, as shown in [10], the distribution of the number of deflections for greedy slotted networks and for unslotted networks offers roughly the same shape (same median, same standard deviation, etc.). This shows that the performance degradation of unslotted routing is mainly due to the difficulty of minimizing the number of deflections locally. This is an algorithmic problem rather than an intrinsic problem of the network asynchronism. In particular, the waste of bandwidth due to the variability of the interpacket spaces does not seem to dominate in the performance degradation of unslotted deflection routing compared to slotted networks.

Conclusion: Problems 1 and 2 either do not exist in practice, or do not have a dominant effect on the performance degradation of unslotted deflection routing. The main problem actually comes from the lack of optimization of the satisfactions of the packet-to-link requests, leading to too many deflections. In the next section, we show how this last problem can be efficiently overcome.

V. A HEURISTIC FOR ROUTING IN UNSLOTTED NETWORKS

Recall that unslotted networks make use of delay loops to detect future arrival of packets at a node, in order to allow or forbid injections of packets at this node. We will make use of these loops for an additional purpose: they will allow us to detect the possible contentions between packets, and thus to maximize the satisfaction of the packet-to-link requests.

A. The Maximum Assignment Problem

Assume that the loops used for inserting packets in unslotted networks produce a delay Δt [see Fig. 1(a)]. The routing decision taken on a packet at time t can benefit from the knowledge of all the future arrivals of packets between times t and $t + \Delta t$ since these packets have been detected by the RCP at the entrances of the delay-loops. Let us consider a packet m_0 routed at time t , and assume that k packets $m_i, i = 1, \dots, k$, will arrive within Δt times. One does not want to maximize the satisfaction of m_0 only, but rather to maximize the global satisfaction of all the $k+1$ packets. For that purpose, we have to take into account the preferences of these packets, and the possible contentions between these packets. Let us denote a packet by a couple (t_1, t_2) where t_1 denotes the arrival time of the packet in the router, and $t_2 - t_1$ denotes its length.

Definition 1: There is a *conflict* between two packets (t_1, t_2) and (t'_1, t'_2) if and only if $t_1 \leq t'_1 \leq t_2$ or $t'_1 \leq t_1 \leq t'_2$. The *conflict graph* is a graph (V, E) where V denotes the set of incoming packets $m_i, i = 0, \dots, k$, and E denotes the set of conflicts between these packets.

Note that the conflict graph is an interval graph [15]. Note also that this graph contains information on the future but not on the past of the current router at the current time. Indeed, none of the packets currently routed are considered in the conflict graph. This notion is captured by another structure: the *preference graph*. Recall that we are currently considering time t . Let $p_{i,j}$ be the preferences of packet m_i for link j . Let s_j be the time at which the output link j will be freed by the packet

currently using link j ($s_j = t - 1$ if no packet is using the link j at time t). A packet (t_1, t_2) will not be allowed to request link j if $s_j > t_1$.

Definition 2: The *preference graph* is a weighted bipartite graph $G = (V_1, V_2, E)$ where V_1 denotes the set of the incoming packets, V_2 denotes the set of the output links, and there is an edge between a packet $m_i = (t_1^{(i)}, t_2^{(i)}) \in V_1$ and a link $j \in V_2$ if and only if $s_j \leq t_1^{(i)}$. An edge between packet m_i and link j has the weight $p_{i,j}$.

Both the conflict graph and the preference graph allow to define the maximum assignment problem. For that purpose, let us formally define what an assignment is:

Definition 3: An *assignment* is a function ϕ from $\{0, \dots, k\}$ to $\{1, \dots, n\}$ such that **1**) if $\phi(i) = j$ then (i, j) is an edge of the preference graph, and **2**) if $\phi(i) = \phi(i')$ then (i, i') is not an edge of the conflict graph.

We aim to solve the following problem, called the *Maximum Assignment Problem* (MA):

$$\text{Finding the assignment } \phi \text{ which maximizes } \sum_{i \in \{0, \dots, k\}} p_{i, \phi(i)}.$$

Solving the maximum assignment problem is NP-complete (see Section VI). Thus, the next section is devoted to a heuristic for the maximum assignment problem.

B. A Heuristic for the Maximum Assignment Problem

1) Description of the Heuristic: The maximum assignment problem is polynomial in slotted networks for two reasons: the conflict graph is the complete graph in this context, and there are at least as many output links as the number of routed packets. The idea of our heuristic consists of simplifying the general problem in unslotted networks in order to get a situation similar to the one in slotted networks. This will allow us to use the standard routing algorithms devoted to slotted networks. Our simplification is based on the fact that, in general, the routing decision for a packet should take more care of the packets arriving soon than of packets arriving much later. Let us formalize this idea.

We use the same notations as in Section V-A, that is m_0 has to be routed at time t . Let n' be the number of free output links at time t . Let $m_1, \dots, m_{k'}, k' \leq k$, be the arriving packets (ordered by their arrival time) that are in conflict with m_0 . Note that two such packets are not necessarily in conflict between themselves, but each is in conflict with m_0 . To route m_0 , our heuristic takes into account m_0 and the $r = \min\{k', n' - 1\}$ next arriving packets. According to what happens for slotted routing, we assume that all these $r+1$ packets are pairwise in conflict. This makes the conflict graph complete. We take, as the preference graph G' of our restricted problem, the subgraph of the original preference graph G induced by the n' output links, and the $r+1$ packets.

Remark: G' is a complete bipartite graph whose two partition sets are the $r+1$ packets on one side, and the n' free output links at time t on the other side. The edge weights are set according to the preferences of the packets for the n' output links.

We get a problem similar to the assignment problem presented in Section II-B for slotted networks. It can be solved

by a polynomial-time maximum-weight matching algorithm in the preference graph G' . (If not perfect, the maximum weight matching in G' can be completed to get a perfect matching by adding edges of weight 0.) Every considered packet, that is each of the next $r + 1$ arriving packets, is assigned to an output link according to this matching. Therefore the current packet can be routed, and this routing avoids many deflections, as far as the r next packets are concerned.

2) *Property of our Heuristic:* The complexity of our heuristic is similar to the complexity of minimizing the number of deflections in slotted networks. Indeed, the complexity of our algorithm is dominated by the search of a maximum weight matching in a bipartite graph, as in the slotted case. Even if such a search is time consuming, an efficient linear heuristic have been proposed in [25]. This heuristic can be applied to efficiently implement the search for a maximum weight matching for 4×4 switches.

Note however that a run of our heuristic is *a priori* required for each packet entering the router. If the difference between the arrival times of two packets m and m' is too small, it is possible that the assignment of the first packet m is not completed before the assignment process of the second packet m' should start. A solution to this problem is to assign m' to the output link specified by our heuristic applied on m . This solution is however not suitable to the case where the heuristic determines the assignment for m_0 and the r next arriving packets m_1, \dots, m_r , and a $r + 1$ th packet arrives before our heuristic completes. As a solution, packets m_0, m_1, \dots, m_r are routed according to the assignment of our heuristic, and m_{r+1} is routed according to its preference as in the standard unslotted deflection routing. This stays true if more than a single packet arrive too early. Actually, it is difficult to evaluate the influence of this phenomenon since it depends on many architectural parameters such as the computational power of the RCP, the number of input links, the distribution of the packet size, etc. In [20], the authors estimated that, with the current technology, the time to compute a maximum matching by a technique similar the one we are using is roughly 10 ns for a bipartite graph 32×32 . At a rate of 625 Mb/s, a time of 10 ns correspond to less than a byte, that is much smaller than the size of the header. Therefore, the probability that the computation for a packet does not complete before the arrival time of another packet is very small.

Although we will see that our heuristic performs quite efficiently, it is of course not optimal as the following example shows. Let us consider a 2×2 switch with two input and two output links, called *North* and *South*. Assume that a long packet preferring the north output link arrives at the same time as a short packet preferring the south output link. Our heuristic will satisfy both packet-to-link requests. However, if a sequence of packets arrive just after the short packet, and if all these packets prefer the north link, then since the large packet is currently routed on the north link, this sequence of packets will be deflected. As far as the maximum assignment problem is concerned, it would have been more efficient to deflect the large packet and the short packet, and to route all the other packets according to their preference. Of course, such a situation rarely occurs in practice, in particular because the average interpacket time is relatively large compared to the average packet size.

C. Experimental Results

Fig. 8(a) presents the throughput of the optimized deflection routing in unslotted networks. As we can check on the figure, the performance increases of about 35% in comparison with the standard unslotted routing. In the experimental context considered in this paper, it enhances unslotted routing almost at the same level as slotted routing.

The link occupation [Fig. 8(b)] is not larger than the link occupation of the non optimized unslotted routing [Fig. 6(b)]. Therefore, 0.7 seems to be the “probabilistic saturation level” of the unslotted routing under the experimental context of this paper. In any case, the optimization of the unslotted routing does not allow to reduce the interpacket space. In some sense, it is a good news since it reduces the dependencies between packets.

Fig. 8(c) shows that our heuristic allows to strongly reduce the average number of packet deflections. As far as the number of deflections is concerned, it makes unslotted routing almost as good as slotted routing. The difference between these two modes is not only a consequence of an approximated solution of an NP-complete problem because even an optimal solution would not have eliminated the fact that: 1) delay loops are of bounded length, and 2) the dependency chain between packets can be very long.

VI. NP-COMPLETENESS OF THE MAXIMUM ASSIGNMENT PROBLEM

In this section, we show that the maximum assignment problem is NP-complete.

Remark: The extended version of the maximum assignment problem in which the conflict graph is arbitrary is trivially NP-complete. Indeed, this problem can be easily reduced the maximum-clique problem, which is NP-complete [14]. The transformation to the maximum-clique problem cannot be applied in the context of this paper because the conflict graphs are interval graphs (for which the maximum-clique problem is polynomial [15]).

We prove the following:

Theorem 1: The following problem is NP-complete:

MAXIMUM ASSIGNMENT (MA)

Instance: A set of n output links, a set of $k + 1$ packets (together with their arrival-times, lengths, and preferences $p_{i,j}$, $i = 0, \dots, k, j = 1, \dots, n$), and an integer K ;

Question: Does there exist an assignment ϕ of these $k + 1$ packets to the n output links such that $\sum_{i \in \{0, \dots, k\}} p_{i, \phi(i)} \geq K$?

Sketch of the Proof: The proof of the NP-completeness is by transformation from the Maximum 2-Satisfiability problem, which has been proved to be NP-complete in [14]:

MAXIMUM 2-SATISFIABILITY (MAX-2SAT)

Instance: a set U of variables, a collection C of clauses over U such that each clause c in C contains two literals, and an integer K' .

Question: Is there a truth assignment for U which simultaneously satisfies at least K' of the clauses in C ?

Roughly speaking, given an instance (U, C, K') of MAX-2SAT, we construct an instance \mathcal{I} of MA as follows. There is a sequence of packets $m_{i,x}$, $i \geq 1$, and a sequence of packets $m_{i,\bar{x}}$, $i \geq 1$, for every variable x . These packets are called

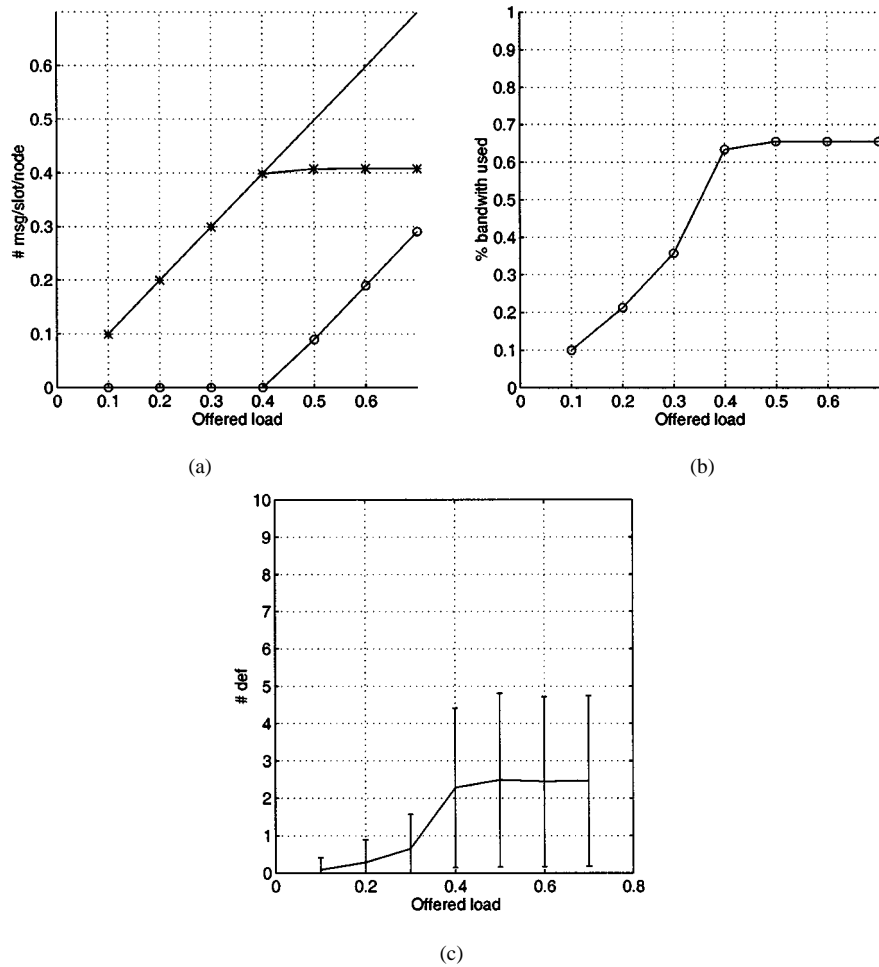


Fig. 8. Throughput, link utilization and average number of deflections of optimized unslotted routing. (a) Throughput (*) and lost packets (o). (b) Link occupation. (c) Average number of deflections.

variable-packets. There are also two packets M_u and M_v for every clause (u, v) . These packets are called *clause-packets*. Two output links play an important role in our setting: if x is true, then all packets $m_{i,x}$ are routed through link 1, otherwise they are all routed through link 2. There are three levels of preferences (that is three different weights): 1, ρ , and ρ^2 . There are additional clause-packets which force the maximum assignment for \mathcal{I} to satisfy the following: the number of clauses that are simultaneously satisfied in (U, C, K') is large if and only if the number of edges of weight 1 in an assignment of maximum weight for \mathcal{I} is large.

The formal proof below explains this correspondence.

Proof: MA is clearly in NP since we can check in polynomial time whether a correct assignment has a weight at least K . Let us describe a polynomial-time transformation of any instance of MAX-2SAT (with no clause of type (x, \bar{x}) or of type (x, x))—MAX-2SAT remains NP-complete in this setting) into an instance of MA. Let an arbitrary instance of MAX-2SAT be

- 1) a set U of ν variables x_1, \dots, x_ν ;
- 2) a set C of μ clauses c_1, \dots, c_μ ;
- 3) an integer K' .

Let us construct an instance of MA corresponding to (U, C, K') . See Fig. 9 for an abstract view of the transformation.

There are $2\mu\nu + 6\mu$ packets in total in the system, that is $k+1 = 2\mu\nu + 6\mu$. These packets are either *variable-packets* or *clause-packets*. They are described hereafter.

The switch has $2\mu + 5$ input (and output) links, labeled from 1 to $2\mu + 5$ ($n = 2\mu + 5$). Two input links are dedicated to the variable-packets. The others links are used by clause-packets.

Variable-Packets: Each variable x in U is represented by a set \mathcal{P}_x of 2μ packets, among which μ packets denoted by $m_{i,x}$, $i = 1, \dots, \mu$, correspond to the literal x , and μ other packets denoted by $m_{i,\bar{x}}$, $i = 1, \dots, \mu$, correspond to the literal \bar{x} . That is

$$\mathcal{P}_x = \{m_{i,x}, m_{i,\bar{x}}, i = 1, \dots, \mu\}.$$

The arrival time t_{i,x_j} of the i th packet of the literal x_j , $j \in \{1, \dots, \nu\}$, satisfies

$$t_{i,x_j} = 20j\mu + 16(i-1).$$

This setting looks rather complicated, but it is simply to avoid contention between two variable-packets corresponding to two different variables. The arrival time t_{i,\bar{x}_j} of the i th packet of the literal \bar{x}_j , $j \in \{1, \dots, \nu\}$, satisfies

$$t_{i,\bar{x}_j} = t_{i,x_j} + 8.$$

All variable-packets are supposed to be of the same length 10.

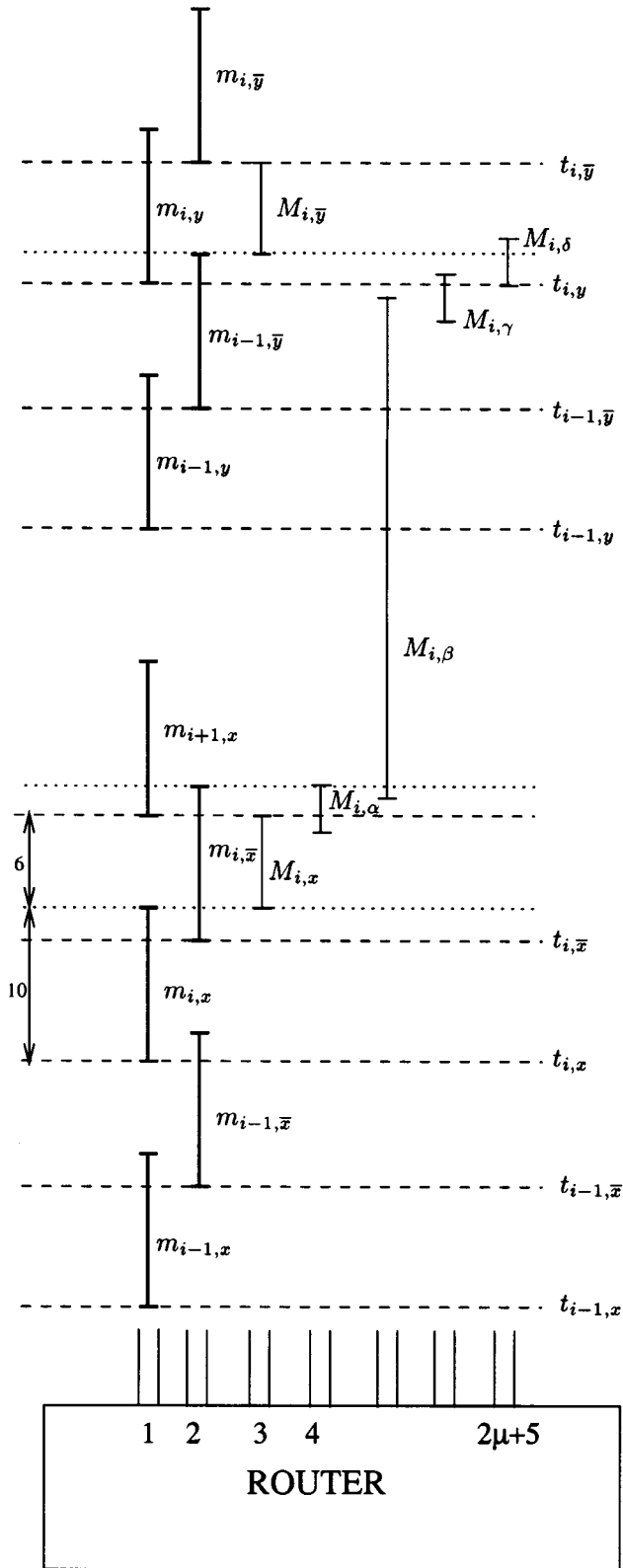


Fig. 9. An example of the transformation of MA into MAX-2SAT in the proof of Theorem 1. In this example, x, y are two variables of U , and $c_i = (x, \bar{y})$ is a clause of C .

The $\mu\nu$ variable-packets $m_{i,x}$ arrive by the input link 1, for all i , and all x . The $\mu\nu$ packets $m_{i,\bar{x}}$ arrive by the input link 2, for all i , and all x . From this setting, packet $m_{1,x_{j+1}}$ arrives

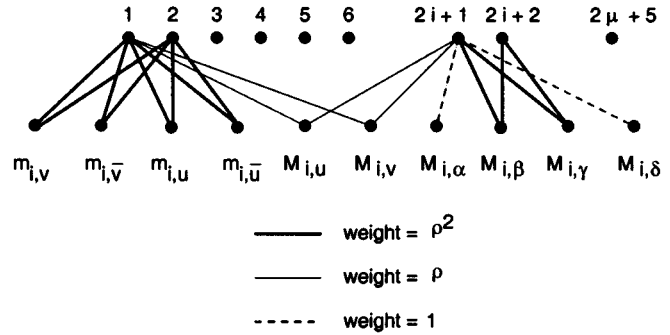


Fig. 10. Abstract view of the preference graph in the proof of Theorem 1. The clause (u, v) belongs to set C .

$4\mu + 16$ tops after packet m_{μ,x_j} . Therefore, there are $4\mu - 2 \geq 2$ time-units between the end of m_{μ,\bar{x}_j} and the beginning of $m_{1,x_{j+1}}$.

Clause-Packets: Let us denote each clause c_i by a couple of literals (u, v) , where $t_{i,u} < t_{i,v}$. Each clause $c_i = (u, v)$ in C , $1 \leq i \leq \mu$, is represented by 6 packets denoted by $M_{i,u}$, $M_{i,v}$, and $M_{i,j}$, $j = \alpha, \beta, \gamma, \delta$. The arrival times T and the lengths L of these packets are set as shown at the bottom of the next page.

The 2μ clause-packets $M_{i,u}$ and $M_{i,v}$, $1 \leq i \leq \mu$, arrive on the input link 3. Since there is no clause of type (x, x) or of type (x, \bar{x}) , this setting is valid. The μ packets of type α arrive on the input link 4. Each of the μ packets of type β arrives on a different input link. These links are labeled from 5 to $\mu + 4$. Similarly, every packet of type γ arrives on a different input link. These links are labeled from $\mu + 5$ to $2\mu + 4$. Finally, the μ packets of type δ arrive on the input link labeled $2\mu + 5$.

Remark: In the structure of our proof, packets $M_{i,u}$ and $M_{i,v}$ play the same role. Similarly, packets $M_{i,\alpha}$ and $M_{i,\delta}$ play the same role. Although packets $M_{i,\beta}$ and $M_{i,\gamma}$ look different, they actually play the same role (to get a totally symmetric situation, it would be possible to balance the lengths of packet $M_{i,\beta}$ and packet $M_{i,\gamma}$, but this would force to control rounding effects.).

Preference Graph: Let $\rho = 2\mu\nu + 6\mu + 1$, that is the total number of packets, plus 1. The preference graph is set as follows (and is summarized in Fig. 10). Recall that it is a complete weighted bipartite graph between the packets and the links.

- 1) For each literal u , the preference of the i th packet $m_{i,u}$, and of the i th packet $m_{i,\bar{u}}$, $i = 1, \dots, \mu$, for the output links 1 and 2 are set to ρ^2 .
- 2) The preferences of $M_{i,\beta}$ and $M_{i,\gamma}$ for the output links $2i + 1$ and $2i + 2$ are set to ρ^2 .
- 3) For each clause $c_i = (u, v)$ in C , $1 \leq i \leq \mu$, the preferences of packets $M_{i,u}$ and $M_{i,v}$ for the output links 1 and $2i + 1$ are set to ρ .
- 4) The preferences of $M_{i,\alpha}$ and $M_{i,\delta}$ for the output link $2i + 1$ are set to 1.
- 5) All the other preferences are set to 0.

Note that only $2\mu + 2$ output links among the $2\mu + 5$ output links of the router receive a non zero preference from a packet.

Objective Value: We set $K = 2\mu(\nu + 1)\rho^2 + 2\mu\rho + K'$.

In order to prove the equivalence of MAX-2SAT and MA, we list below properties that must be satisfied by the maximum

assignment of an instance of MA obtained from an instance of MAX-2SAT.

Lemma 1: Any assignment of value at least K involves $2\mu(\nu + 1)$ edges of weight ρ^2 , and 2μ edges of weight ρ .

Proof: Let ϕ be an assignment of weight at least K . Assume that in ϕ , there are a_2 edges of weight ρ^2 , a_1 edges of weight ρ , and a_0 edges of weight 1. If $a_2 < 2\nu\mu + 2\mu$, then $a_0 + a_1\rho \geq \rho^2$. This is in contradiction with the definition of ρ , and with the fact that $a_0 + a_1$ is at most equal to the number of packets. Thus ϕ contains at least $2\nu\mu + 2\mu$ edges of weight ρ^2 . There are only $2\nu\mu + 2\mu$ packets incident to the edges of weight ρ^2 . Therefore ϕ contains exactly $2\mu(\nu + 1)$ edges of weight ρ^2 . We apply the same argument for the edges of weight ρ to show that the assignment ϕ contains exactly 2μ edges of weight ρ . ■

As a direct consequence of the previous lemma, we get:

Lemma 2: In any assignment of value at least K , if a packet m is incident to an edge of weight ρ (resp. ρ^2) in the preference graph, then the assignment contains an edge incident to m that has a weight at least ρ (resp. ρ^2).

As a third property, we have:

Lemma 3: In any assignment ϕ of value at least K , for every x , all the packets $m_{i,x}$, $i \in \{1, \dots, \mu\}$, must be assigned to the same output link, which can only be link 1 or link 2. The same property holds for the packets $m_{i,\bar{x}}$, $i \in \{1, \dots, \mu\}$. Moreover, for every x , packets $m_{i,x}$, $i \in \{1, \dots, \mu\}$, on one hand, and packets $m_{i,\bar{x}}$, $i \in \{1, \dots, \mu\}$, on the other hand, are not assigned to the same link.

Proof: By Lemma 2, packets $m_{i,x}$ and $m_{i,\bar{x}}$ are assigned to link 1 or 2.

Assume first that the assignment ϕ contains the edge $(m_{1,x}, 1)$. Since there is a conflict between $m_{1,x}$ and $m_{1,\bar{x}}$, the edge $(m_{1,\bar{x}}, 2)$ is in ϕ . Since there is a conflict between $m_{1,\bar{x}}$ and $m_{2,x}$, the edge $(m_{2,x}, 1)$ is in ϕ . This argument applies successively for all i 's and the lemma holds.

The case in which the assignment ϕ contains the edge $(m_{1,\bar{x}}, 1)$ yields the same result. ■

Based of the previous lemmas, let us show that there exists a truth assignment for the variables in U such that at least K' of the clauses in C are simultaneously satisfied if and only if there exists an assignment ϕ of weight at least K .

Sufficient Condition: Assume that there exists an assignment ϕ of weight at least K . Let us construct a truth assignment τ for (U, C, K') . For every $x \in U$, we set $\tau(x)$ as follows:

$$(m_{1,x}, 1) \in \phi \Leftrightarrow \tau(x) = \text{true}.$$

Let us count the number of clauses that are simultaneously satisfied by τ . Assume that the clause $c_i = (u, v)$ is not satisfied by τ . Then both $(m_{j,\bar{u}}, 1)$ and $(m_{j,\bar{v}}, 1)$ are in ϕ for every j . Since packets $m_{i,\bar{u}}$ and $M_{i,u}$ are in conflict, the edge $(M_{i,u}, 1)$ is not in ϕ . Since this edge is of weight ρ , Lemma 2 implies that

the edge $(M_{i,u}, 2i + 1)$ is in ϕ . Moreover, since packets $M_{i,u}$ and $M_{i,\alpha}$ are in conflict, the edge $(M_{i,\alpha}, 2i + 1)$ is not in ϕ . Similarly, the edge $(M_{i,\delta}, 2i + 1)$ is not in ϕ . Thus, the weights of the two edges corresponding to the assignment of the packets $M_{i,\alpha}$ and $M_{i,\delta}$ are both equal to zero.

Moreover, thanks to Lemma 2, $(M_{i,\alpha}, 2i + 1)$ and $(M_{i,\delta}, 2i + 1)$ cannot be both in ϕ because we must have either $(M_{i,\beta}, 2i + 1) \in \phi$ and $(M_{i,\gamma}, 2i + 2) \in \phi$, or $(M_{i,\beta}, 2i + 2) \in \phi$ and $(M_{i,\gamma}, 2i + 1) \in \phi$.

Therefore, the number of clauses satisfied by τ is at least the number of edges of weight 1 in ϕ .

Let a_0 denotes the number of edges of weight 1 in ϕ . Since the weight of ϕ is at least K , we have, thanks to Lemma 1:

$$\text{Weight}(\phi) = (2\nu\mu + 2\mu)\rho^2 + 2\mu\rho + a_0.$$

Therefore

$$\text{Weight}(\phi) \geq K = (2\nu\mu + 2\mu)\rho^2 + 2\mu\rho + K'$$

implies that $a_0 \geq K'$. Therefore, the truth assignment τ for U simultaneously satisfies at least K' of the clauses in C .

Necessary Condition: Assume that there exists a truth assignment τ for U that simultaneously satisfies at least K' of the clauses in C . We construct an assignment ϕ as follows. For every $i = 1, \dots, \mu$:

- 1) For every x , if $\tau(x) = \text{true}$ then $(m_{i,x}, 1) \in \phi$ and $(m_{i,\bar{x}}, 2) \in \phi$, otherwise $(m_{i,x}, 2) \in \phi$ and $(m_{i,\bar{x}}, 1) \in \phi$.
- 2) If $c_i = (u, v)$ is not satisfied, then $(M_{i,u}, 2i + 1)$, $(M_{i,v}, 2i + 1)$, $(M_{i,\beta}, 2i + 1)$, and $(M_{i,\gamma}, 2i + 2)$ are in ϕ , otherwise two cases:

Case 1: Only one of the two literals u and v is true.

- If u is true, then $(M_{i,u}, 1)$, $(M_{i,v}, 2i + 1)$, $(M_{i,\alpha}, 2i + 1)$, $(M_{i,\beta}, 2i + 2)$, and $(M_{i,\gamma}, 2i + 1)$ are in ϕ .
- If v is true, then $(M_{i,u}, 2i + 1)$, $(M_{i,v}, 1)$, $(M_{i,\delta}, 2i + 1)$, $(M_{i,\gamma}, 2i + 2)$, and $(M_{i,\beta}, 2i + 1)$ are in ϕ .

Case 2: Both literals u and v are true. Then $(M_{i,u}, 1)$, $(M_{i,v}, 1)$, $(M_{i,\alpha}, 2i + 1)$, $(M_{i,\beta}, 2i + 2)$, and $(M_{i,\gamma}, 2i + 1)$ are in ϕ .

Let us count the weight of ϕ . In all cases, $2\mu(\nu + 1)$ edges of weight ρ^2 are in ϕ . Similarly, 2μ edges of weight ρ are also in ϕ , without conflict within themselves or with edges of weight ρ^2 . More importantly, if c_i is satisfied, then either $(M_{i,\alpha}, 2i + 1)$ or $(M_{i,\delta}, 2i + 1)$ is in ϕ . Therefore, at least K' edges of weight 1 are in ϕ , and hence ϕ has a weight at least K .

Conclusion: There exists a truth assignment for U that simultaneously satisfies at least K' of the clauses in C if and only if there exists an assignment ϕ such that its weight is at least K . Therefore MA is NP-complete. ■

$$\begin{array}{llll} T_{i,u} = t_{i,\bar{u}} + 2 & L_{i,u} = 6 & T_{i,v} = t_{i,\bar{v}} + 2 & L_{i,v} = 6 \\ T_{i,\alpha} = t_{i,\bar{u}} + 7 & L_{i,\alpha} = 3 & T_{i,\beta} = t_{i,\bar{u}} + 9 & L_{i,\beta} = t_{i,\bar{v}} - t_{i,\bar{u}} - 10 \\ T_{i,\gamma} = t_{i,\bar{v}} - 2 & L_{i,\gamma} = 3 & T_{i,\delta} = t_{i,\bar{v}} & L_{i,\delta} = 3 \end{array}$$

The following result is a direct consequence of Theorem 1:

Corollary: The following problem is NP-complete:

GENERAL MAXIMUM ASSIGNMENT (GMA)

Instance: An interval graph $G = (V, E)$, a weighted bipartite graph $H = (V_1, V_2, F)$ where $V_1 = V$, and an integer K ;

Question: Does there exist a subset $\phi \subset F$ such that 1) every vertex of V_1 is the extremity of exactly one edge in ϕ , 2) if two edges of ϕ are incident to the same vertex of V_2 , say $e = (v_1, v_2)$ and $e' = (v'_1, v_2)$, then the edge $(v_1, v'_1) \notin E$, and 3) the sum of the weights of the edges in ϕ is $\geq K$?

VII. CONCLUSION

The two main contributions of this paper are:

- 1) The performances degradation of unslotted deflection routing, compared to slotted deflection routing, is mostly due to the fact that slotted networks naturally allow a global optimization of the packet-to-link requests, whereas unslotted systems treat requests one by one. In particular
 - a) no congestion phenomenons were observed in our simulations;
 - b) even if a waste of bandwidth due to the variability of the interpacket spaces in unslotted network was observed, it is not the major reason for the decrease of the throughput.
- 2) It is possible to provide a simple heuristic for a global optimization of the packet-to-link requests in unslotted systems. This heuristic does not require additional hardware. It just makes use of the delay-loops present in most unslotted systems. Our simulations have shown that our heuristic allows to enhance the performances of unslotted systems almost at the same level as slotted systems.

As a consequence, we have shown that unslotted deflection routing can be implemented in a way which makes it a competitive alternative to slotted deflection routing for optical time division multiplexing deflection networks. We are currently exploring different extensions of this work, in particular by searching for a good way to mix WDM and TDM in the context of deflection routing.

APPENDIX A TIME SCALING

We have considered two time scaling in order to separate the behavior of the network from the behavior of the applications using the network. The main purpose of these two ticks is to perform simulation on synchronous and asynchronous networks using exactly the same probabilistic law for the emission process. We have considered:

- 1) simulation tick, or network tick;
- 2) processor tick, or emission tick.

At every simulation tick, we consider possible emission of packets at each node, and we route packets in the network. The traffic demand is simulated as follows. At each node, the decision to inject or not a packet in the input queue is taken according to a probabilistic law, and destinations are chosen uniformly at random. Each node follows the same law. At each processor, the emission follows a Bernoulli law (when a processor sends,

it sends exactly one packet). This Bernoulli law is in turn simulated by a Binomial law at the simulation tick. We denote by t_n (resp. t_p), the tick of the network (resp. of the processor). In our experiments, we have fixed $t_n = 0.1 \mu s$, and $t_p = 20 ns$. Note that other experiments done with a smaller simulation tick ($t_n = 20 ns$) produced the same results.

Most of our experimental results are presented as a function of the load offered to the network. The offered load is expressed in packets per node and per slot. (In unslotted networks, the slot is an abstract measure expressing the maximum size of a packet.) The time slot is denoted by t_s . We have fixed the time slot at $t_s = 1 \mu s$. Hence, to get a fixed offered load \mathcal{L} , we have forced the parameter of the Bernoulli law $B(\lambda)$ of the emissions to be $\lambda = \mathcal{L}/(t_s/t_p)$. Therefore, the emission law of the network is $B(\lambda, t_n/t_p)$. This protocol produces the same emission law for both slotted and unslotted networks. Note that it would not have been the case if we would have followed the naive approach consisting of setting $t_n = t_s$ for synchronous simulation.

APPENDIX B SPORADIC TRAFFIC

We mainly consider two different emission laws for two different kinds of experiments: Poisson traffic, and sporadic traffic. In the Poisson traffic, every processor follows the same Bernoulli law. This is the most commonly studied traffic in the literature.

In order to simulate a sporadic traffic, we have used an emission law denoted by

$$S(\mathcal{L}_g, p, \mathcal{L}_b, p').$$

This law is a two states Markovian chain. More precisely, each node is in two possible states called *ground* and *bursty*. These states alternate according to two probabilities p and p' . From the ground state, the probability to enter the bursty state is p . From the bursty state, the probability to enter the ground state is p' . In the ground state, the emission law is Poisson. In the bursty state, we allow processors to send a large number of packets within one slot (such packets will be stored in the input queue). When a processor is in the bursty state, its offered load is of average $\mathcal{L}_b > 1$ (to be compared with the global offered load in the Poisson traffic which is always strictly less than 1).

As in [21], we have considered that bursty traffics are mainly caused by ftp-data-like applications. Moreover, whatever the load of the network is, a burst offers the same characteristic. Thus, we have set $\mathcal{L}_b = cst$, independently from the global load \mathcal{L} . For the same reasons, the probability p' to get out of a bursty application is not related to the global load, and thus it is set as a constant. The ground emission rate \mathcal{L}_g is defined as a linear function of the offered load of the network \mathcal{L} . Indeed, the ground traffic is induced by telnet-like connections [21] whose number grows linearly with the number of running applications. We have set $\mathcal{L}_g = c\mathcal{L}$. Note that c should not be larger than the saturation threshold of the network. The constant c is hence set to 0.3. For a given offered load, the probability p is fixed to $p'(\mathcal{L}(1-c)/\mathcal{L}_b - \mathcal{L})$ so that the mean of the law $S(\mathcal{L}_g, p, \mathcal{L}_b, p')$ is \mathcal{L} . Thence, in our sporadic model, an

increase of the load will be induced by an increase of the frequency at which one enters the bursty state.

We have fixed (somewhat arbitrarily) the value of \mathcal{L}_b at 5 (smaller bursts would not be significant, and larger bursts would saturate the input queues). We also set $p' = 0.02$. This value was fixed according to the value of \mathcal{L}_b . It implies that a burst will fill up the input queue with 25 packets on average, that is with 25% of the size of the queue.

ACKNOWLEDGMENT

The authors would like to thank F. Clérot for many helpful comments on many aspects of our research on all-optical networks.

REFERENCES

- [1] A. S. Acampora and M. J. Karol, "An overview of lightwave packet networks," *IEEE Networks*, pp. 29–41, Jan. 1989.
- [2] A. S. Acampora and S. I. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082–1089, June 1992.
- [3] H. Badr and S. Pöder, "An optimal shortest-path routing policy for network computers with regular mesh-connected topologies," *IEEE Trans. Comput.*, vol. 38, pp. 1362–1371, Oct. 1989.
- [4] C. Baransel, W. Dobosiewicz, and P. Gburzynski, "Routing in multihop packet switching: Gb/s challenge," *IEEE Network*, pp. 38–61, May 1995.
- [5] D. J. Blumenthal, P. R. Prucnal, and J. R. Sauer, "Photonic packet switches: Architectures and experimental implementations," *Proc. IEEE*, vol. 82, pp. 1650–1667, Nov. 1994.
- [6] F. Borgonovo and L. Fratta, "Deflection networks: Architectures for metropolitan and wide area networks," *Comput. Networks ISDN Syst.*, vol. 2, pp. 171–183, 1992.
- [7] F. Borgonovo, L. Fratta, and J. Bannister, "Unslotted deflection routing in all-optical networks," in *IEEE Globecom*, 1993, pp. 119–125.
- [8] ———, "On the design of optical deflection-routing networks," in *IEEE Infocom*, 1994, pp. 120–129.
- [9] J. Brassil and R. Cruz, "Bounds on maximum delay in networks with deflection routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 6, pp. 724–732, July 1995.
- [10] T. Chich and P. Fraigniaud, "An extended comparison of slotted and unslotted deflection routing," in *IEEE ICCCN*, 1997, pp. 92–97.
- [11] I. Chlamtac *et al.*, "CORD: Contention resolution by delay lines," *IEEE J. Select. Areas Commun.*, vol. 14, May 1996.
- [12] C. Fang and T. Szymanski, "An analysis of deflection routing multidimensional regular mesh networks," in *IEEE Infocom*, vol. 3, 1991, pp. 859–868.
- [13] J. Fehrer, J. Sauer, and L. Ramfelt, "Design and implementation of a prototype optical deflection network," in *ACM SIGCOMM*, vol. 24, 1994, pp. 191–200.
- [14] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide To The Theory of the NP-Completeness*. San Francisco, CA: Freeman, 1979.
- [15] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*. New York: Academic, 1980.
- [16] Z. Haas, "The 'staggering switch', an electrically controlled optical packet switch," *J. Lightwave Technol.*, vol. 11, May/June 1993.

- [17] H. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Q.*, vol. 2, pp. 83–97, 1955.
- [18] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.
- [19] N. F. Maxemchuk, "Routing in the MSN," *IEEE Trans. Commun.*, vol. 35, pp. 503–512, May 1987.
- [20] A. Mekittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," in *IEEE Infocom*, vol. 2, 1998, pp. 792–799.
- [21] V. Paxson and S. Floyd, "Wide area traffic: The failure of poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–244, June 1995.
- [22] M. Renaud, F. Masetti, C. Guillemot, and B. Bostica, "Network and system concepts for optical packet switching," *IEEE Commun. Mag.*, pp. 96–102, Apr. 1997.
- [23] S.-W. Seo, K. Bergman, and P. R. Prucnal, "Transparent optical networks with time-division multiplexing," *J. Select. Areas Commun.*, vol. 14, pp. 1039–1051, June 1996.
- [24] Z. Wang and J. Crowcroft, "QoS routing for supporting multimedia applications," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1228–1234, July 1996.
- [25] J. S. Wong and Y. Kang, "Distributed and fail-safe routing algorithms in toroidal-based metropolitan area networks," *Comput. Networks ISDN Syst.*, vol. 18, pp. 379–391, 1989/90.



Thierry Chich received the Ph.D. degree in 1997 from the Ecole Normale Supérieure de Lyon.

He is currently Computer Systems Engineer at the University of Clermont-Ferrand, Aubière, France.



Johanne Cohen received the Magistère degree in computer science from the Ecole Normale Supérieure, Lyon, France, and the Ph.D. degree in computer science from Paris South University, Paris, France, in 1995 and 1998, respectively.

Since 1999, she has been a Maître de Conférence at the LORIA Laboratory, Nancy, France. Her research interests include algorithms for network communications.

Pierre Fraigniaud received the Ph.D. degree from the Ecole Normale Supérieure, Lyon, France, in 1990.

He is currently Chargé de Recherche with CNRS, Université Paris-Sud, France. His research interests cover several fundamental aspects of routing and information dissemination in networks.